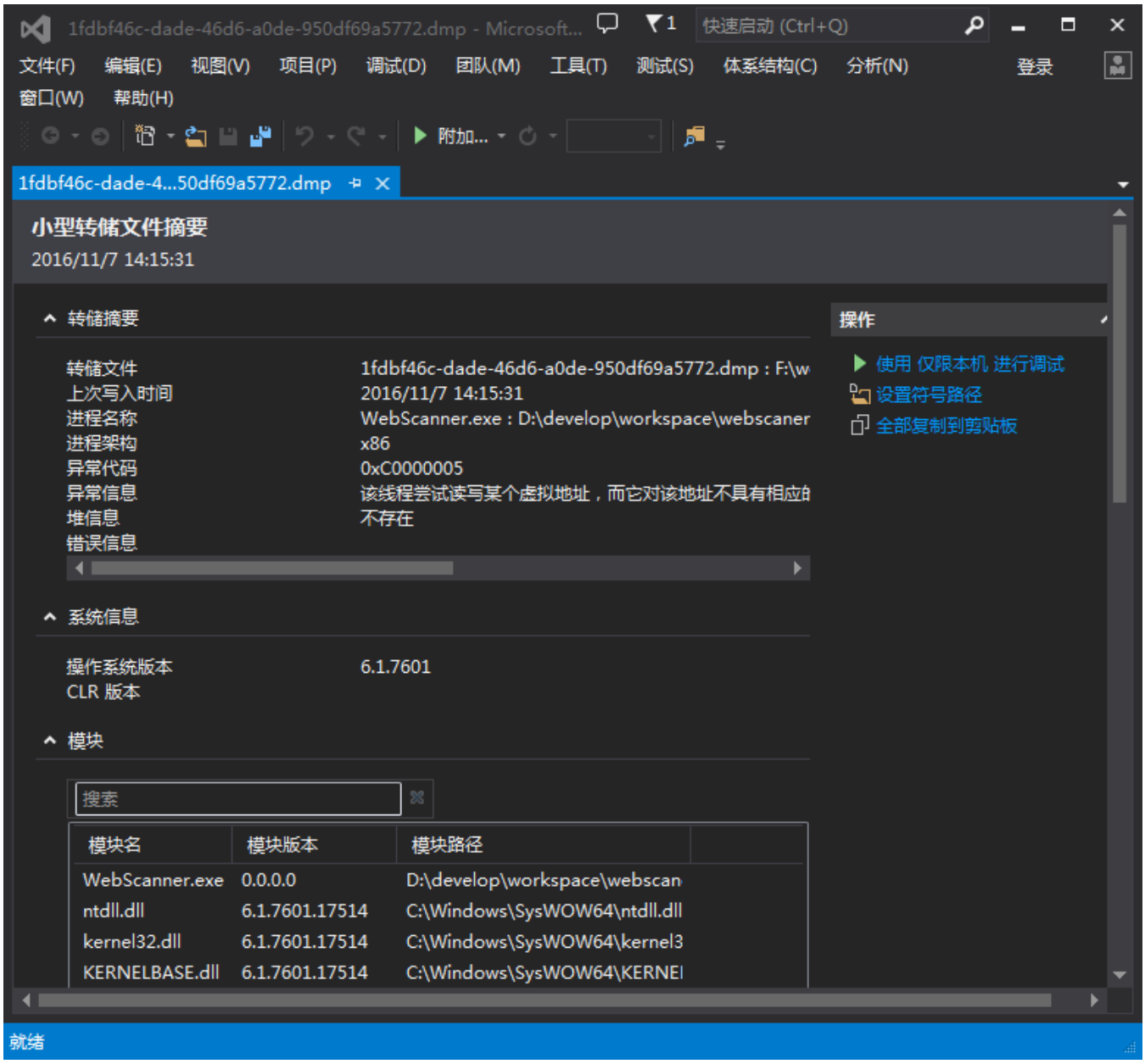
疑难杂症系列：Win 下 dump 定位错误到代码行心得

一．背景

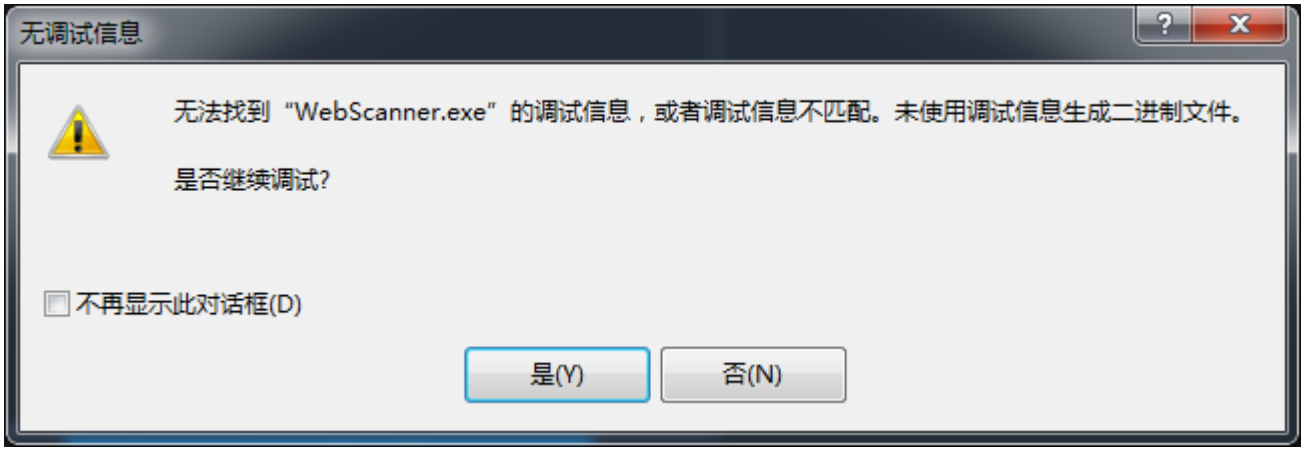
（balabalabalabala）一句话概括：请问只有minidump和目前的源码来定位这个历史版本程序产生的dump出错的代码行/块/片/文件是怎么样的体验?

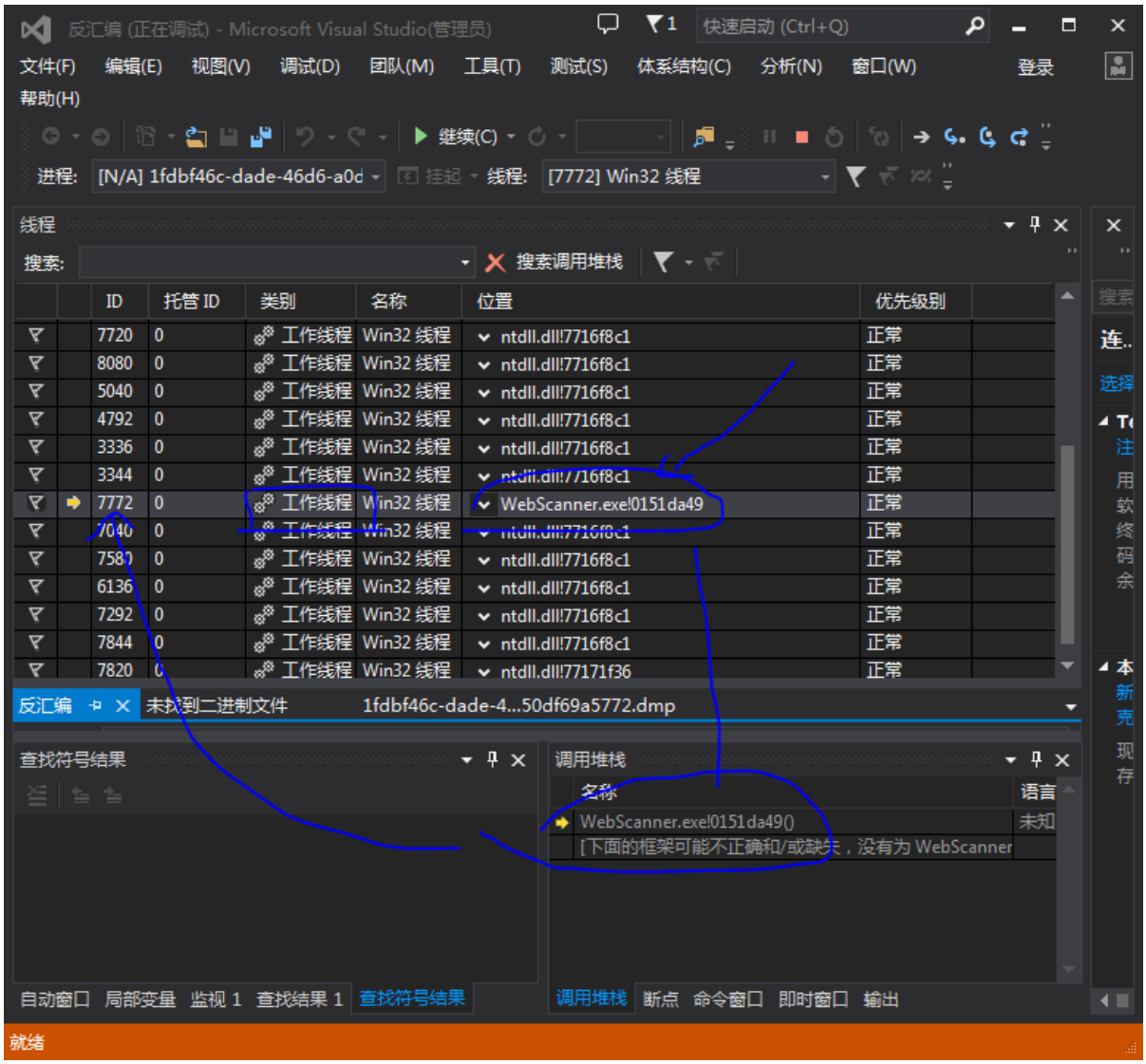
二．实践

选择一款可以打开dump文件的调试器，例如windbg，这次使用的是vs自带的功能，使用vs2013打开dump文件。



由于没有符号和目标的exe文件，我们能够得到的信息很少。





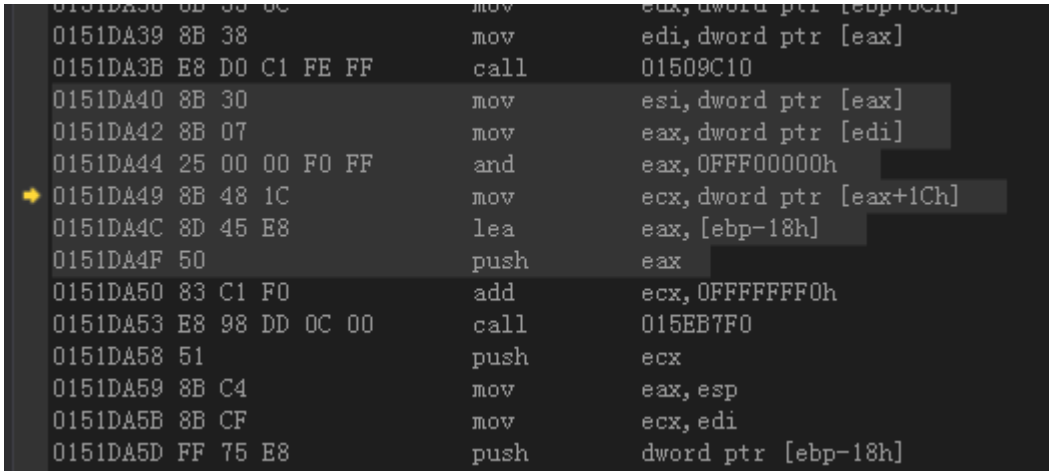
调用堆栈可以得到一个最后的异常地址，以及线程信息中看到奔溃的线程ID和代码空间（在用户空间中crash）。

三．方案

我们有的：

1.在代码段**0x0151DA49**产生异常。

2.异常类型是 Access Violation



3.异常地址的汇编代码若干。

4.我们有目前最新的代码。

5.错误版本的代码和目前最新版本代码改动不大。

具体方案：

由于代码改动不大，那么就意味着重新编译新代码产生的汇编代码也是一样的。故通过新编译的程序中寻找特征代码（crash的机器码）

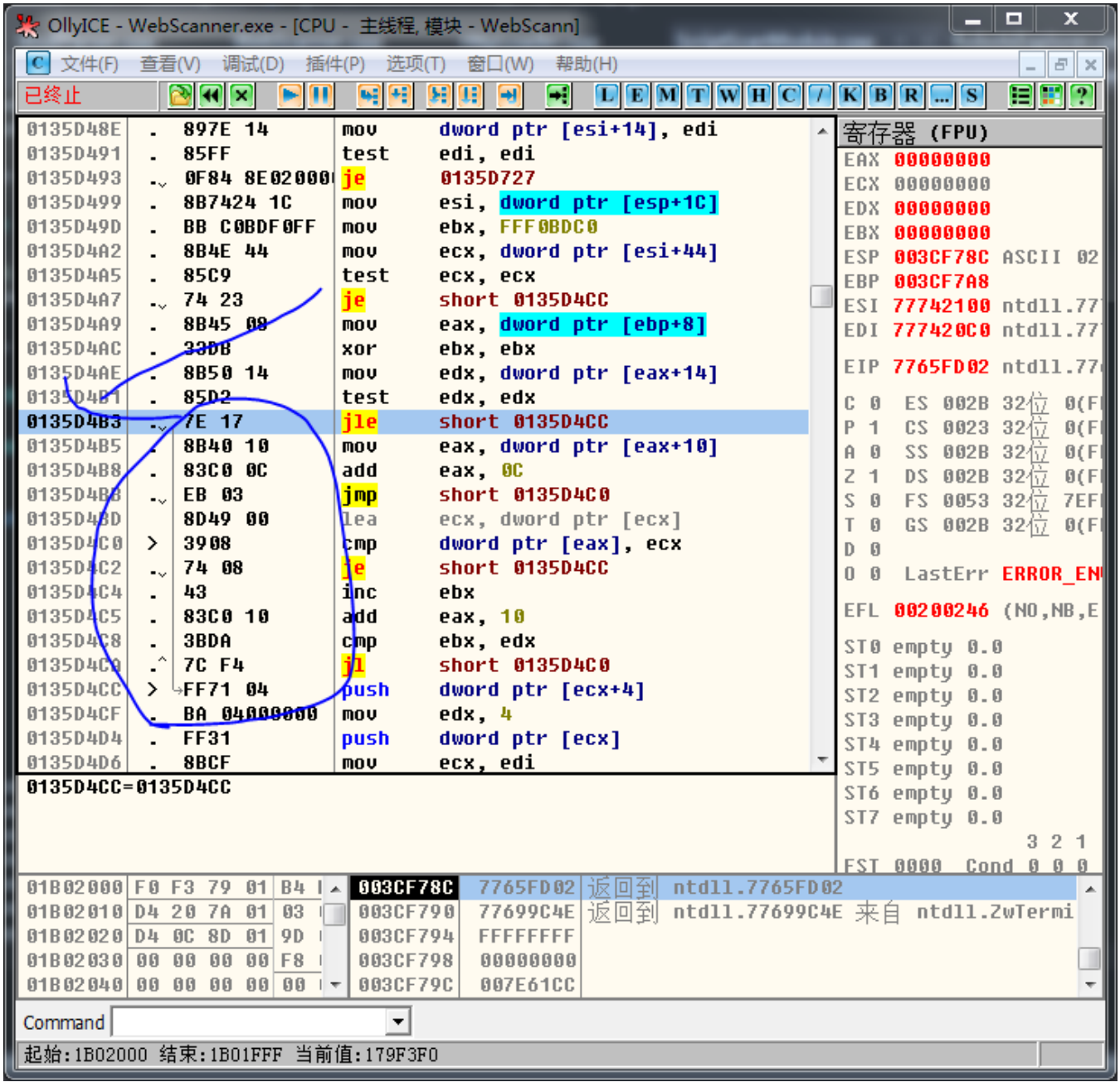
来定位我们的代码行，由于我们有源码新编译程序可以关联pdb和汇编快速定位代码行。

用OD（ollydbg）打开我们新编译的程序（最好关闭ALRS编译，方便OD加载调试），关联pdb和源码。

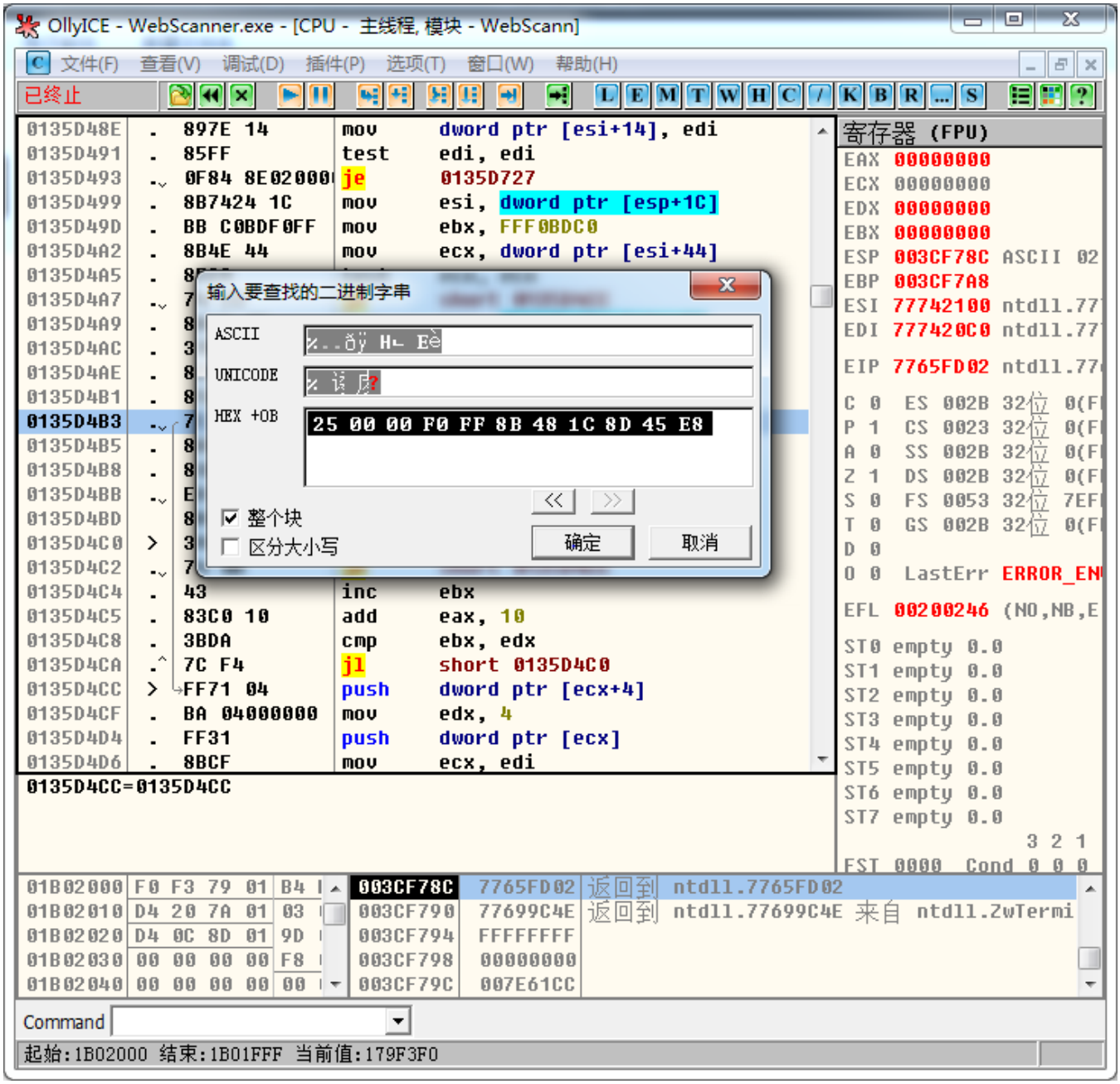
可以发现OD加载完毕之后，已经解析出符合了。

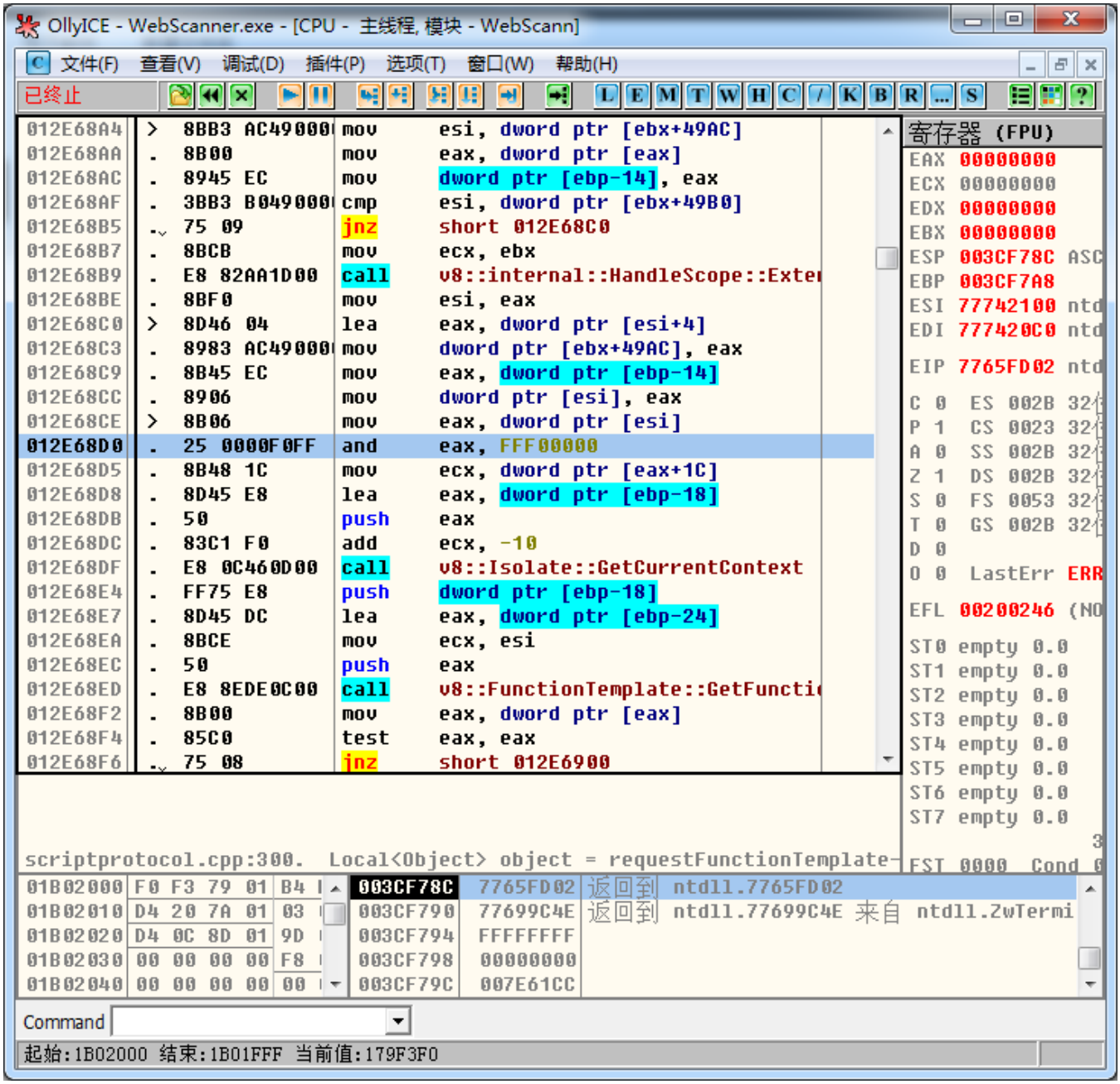
碰下运气，我们去报错的地址看下。

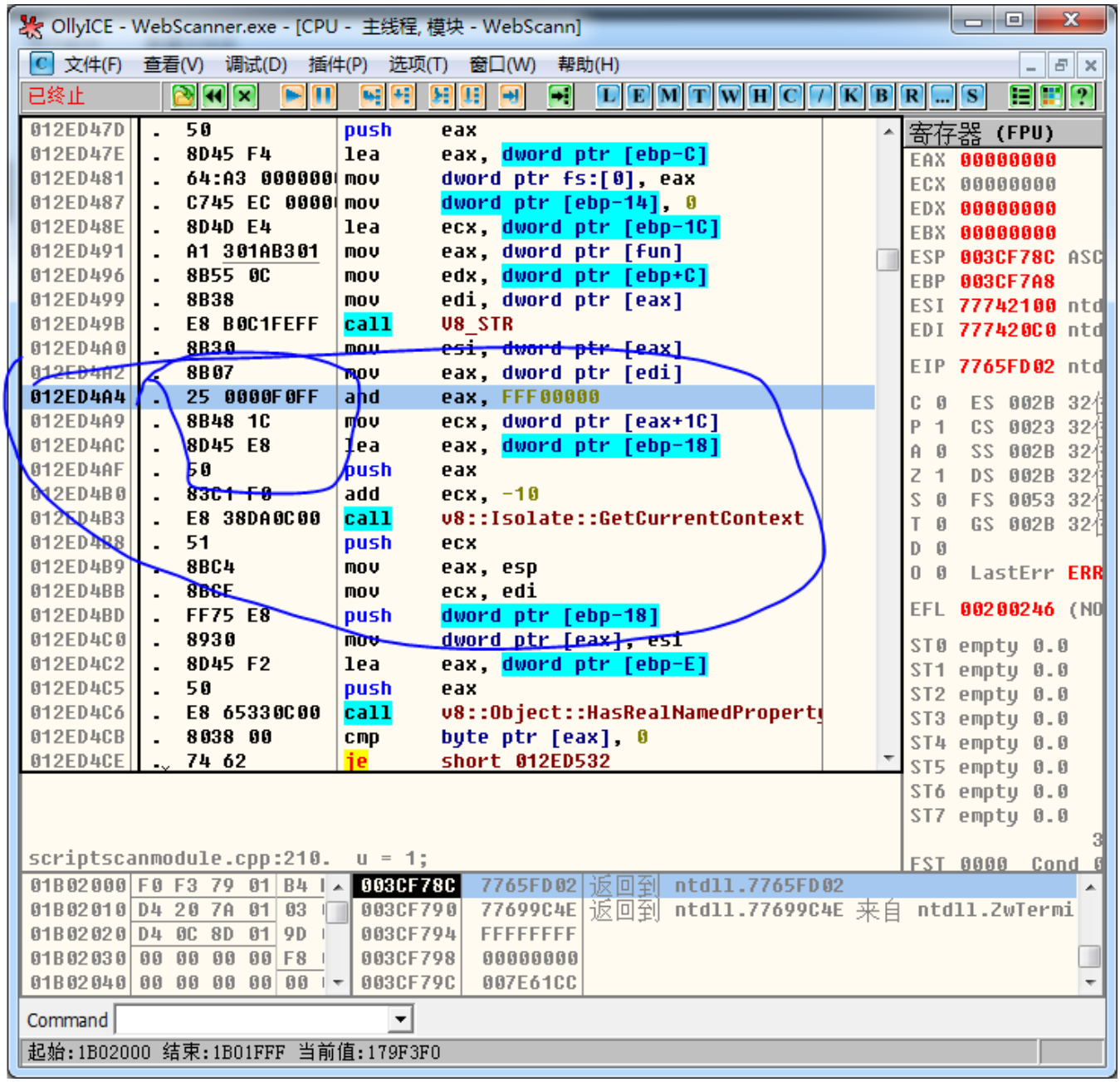
发现新编译程序这个地址空间并没有给我们机会，那么使用方案中。



二进制查找特征码，找到很多处，最后定位到上下文跟DUMP中一致的地方。

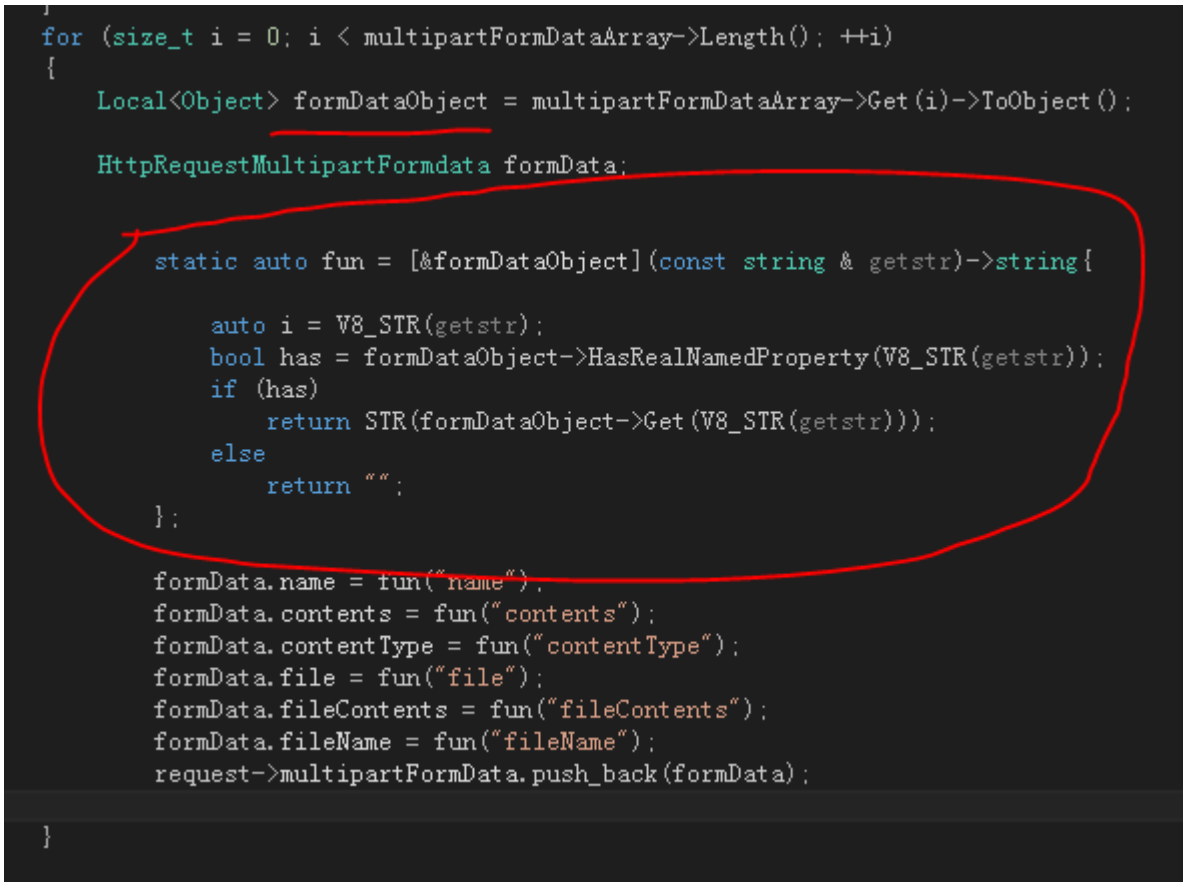

OllyICE - WebScanner.exe - [CPU - 主线程, 模块 - WebScann]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H)
已终止
0135D48E . 897E 14 mov dword ptr [esi+14], edi
0135D491 . 85FF test edi, edi
0135D493 . 0F84 8E02000 je 0135D727
0135D499 . 8B7424 1C mov esi, dword ptr [esp+1C]
0135D49D . BB C0BDF0FF mov ebx, FFF0BDC0
0135D4A2 . 8B4E 44 mov ecx, dword ptr [esi+44]
0135D4B3 . 7
0135D4C0 > 3
0135D4C2 . 7
0135D4C4 . 43 inc ebx
0135D4C5 . 83C0 10 add eax, 10
0135D4C8 . 3BDA cmp ebx, edx
0135D4CA . 7C F4 jl short 0135D4C0
0135D4CC > FF71 04 push dword ptr [ecx+4]
0135D4CF . BA 04000000 mov edx, 4
0135D4D4 . FF31 push dword ptr [ecx]
0135D4D6 . 8BCF mov ecx, edi
0135D4CC=0135D4CC
输入要查找的二进制字串
ASCII
UNICODE
HEX +0B
25 00 00 F0 FF 8B 48 1C 8D 45 E8
整个块
区分大小写
确定
取消
寄存器 (FPU)
EAX 00000000
ECX 00000000
EDX 00000000
EBX 00000000
ESP 003CF78C ASCII 02
EBP 003CF7A8
ESI 77742100 ntdll.77
EDI 777420C0 ntdll.77
EIP 7765FD02 ntdll.77
C 0 ES 002B 32位 0(F
P 1 CS 0023 32位 0(F
A 0 SS 002B 32位 0(F
Z 1 DS 002B 32位 0(F
S 0 FS 0053 32位 7EF
T 0 GS 002B 32位 0(F
D 0
O 0 LastErr ERROR_EN
EFL 00200246 (NO,NB,E
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
3 2 1
FST 0000 Cond 0 0 0
01B02000 F0 F3 79 01 B4
01B02010 D4 20 7A 01 03
01B02020 D4 0C 8D 01 9D
01B02030 00 00 00 00 F8
01B02040 00 00 00 00 00
003CF78C 7765FD02 返回到 ntdll.7765FD02
003CF790 77699C4E 返回到 ntdll.77699C4E 来自 ntdll.ZwTermi
003CF794 FFFFFFFF
003CF798 00000000
003CF79C 007E61CC
Command
起始:1B02000 结束:1B01FFF 当前值:179F3F0


OllyICE - WebScanner.exe - [CPU - 主线程, 模块 - WebScann]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H)
已终止
012E68A4 > 8BB3 AC49000(mov esi, dword ptr [ebx+49AC]
012E68AA . 8B00 mov eax, dword ptr [eax]
012E68AC . 8945 EC mov dword ptr [ebp-14], eax
012E68AF . 3BB3 B049000(cmp esi, dword ptr [ebx+49B0]
012E68B5 . 75 09 jnz short 012E68C0
012E68B7 . 8BCB mov ecx, ebx
012E68B9 . E8 82AA1D00 call v8::internal::HandleScope::Exte
012E68BE . 8BF0 mov esi, eax
012E68C0 > 8D46 04 lea eax, dword ptr [esi+4]
012E68C3 . 8983 AC49000(mov dword ptr [ebx+49AC], eax
012E68C9 . 8B45 EC mov eax, dword ptr [ebp-14]
012E68CC . 8906 mov dword ptr [esi], eax
012E68CE > 8B06 mov eax, dword ptr [esi]
012E68D0 . 25 0000F0FF and eax, FFF00000
012E68D5 . 8B48 1C mov ecx, dword ptr [eax+1C]
012E68D8 . 8D45 E8 lea eax, dword ptr [ebp-18]
012E68DB . 50 push eax
012E68DC . 83C1 F0 add ecx, -10
012E68DF . E8 0C460D00 call v8::Isolate::GetCurrentContext
012E68E4 . FF75 E8 push dword ptr [ebp-18]
012E68E7 . 8D45 DC lea eax, dword ptr [ebp-24]
012E68EA . 8BCE mov ecx, esi
012E68EC . 50 push eax
012E68ED . E8 8EDE0C00 call v8::FunctionTemplate::GetFunctio
012E68F2 . 8B00 mov eax, dword ptr [eax]
012E68F4 . 85C0 test eax, eax
012E68F6 . 75 08 jnz short 012E6900
寄存器 (FPU)
EAX 00000000
ECX 00000000
EDX 00000000
EBX 00000000
ESP 003CF78C ASC
EBP 003CF7A8
ESI 77742100 ntd
EDI 777420C0 ntd
EIP 7765FD02 ntd
C 0 ES 002B 32
P 1 CS 0023 32
A 0 SS 002B 32
Z 1 DS 002B 32
S 0 FS 0053 32
T 0 GS 002B 32
D 0
O 0 LastErr ERR
EFL 00200246 (NO
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FST 0000 Cond
scriptprotocol.cpp:300. Local<Object> object = requestFunctionTemplate-
01B02000 F0 F3 79 01 B4
01B02010 D4 20 7A 01 03
01B02020 D4 0C 8D 01 9D
01B02030 00 00 00 00 F8
01B02040 00 00 00 00 00
003CF78C 7765FD02 返回到 ntdll.7765FD02
003CF790 77699C4E 返回到 ntdll.77699C4E 来自 ntdll.ZwTermi
003CF794 FFFFFFFF
003CF798 00000000
003CF79C 007E61CC
Command
起始:1B02000 结束:1B01FFF 当前值:179F3F0



0x012ED4A4,非常方便，OD已经有符号了，可以直接看出代码行（若没有加载符号可以用别的方法，这里就不赘述了）。

VS中找到出错代码。



```
for (size_t i = 0; i < multipartFormDataArray->Length(); ++i)
{
    Local<Object> formDataObject = multipartFormDataArray->Get(i)->ToObject();

    HttpRequestMultipartFormdata formData;


        static auto fun = [&formDataObject](const string & getstr)->string{

            auto i = V8_STR(getstr);
            bool has = formDataObject->HasRealNamedProperty(V8_STR(getstr));
            if (has)
                return STR(formDataObject->Get(V8_STR(getstr)));
            else
                return "";
        };

        formData.name = fun("name");
        formData.contents = fun("contents");
        formData.contentType = fun("contentType");
        formData.file = fun("file");
        formData.fileContents = fun("fileContents");
        formData.fileName = fun("fileName");
        request->multipartFormData.push_back(formData);

}
```

四：分析问题产生原因

后续补充

另外PS： 单线程不报错，但是这个代码真的对吗?

By luoyue